

Remarks

I. 35 U.S.C. §102

A. The Standard

“Under 35 U.S.C. § 102, every limitation of a claim must identically appear in a single prior art reference for it to anticipate the claim.” *Gechter v. Davidson*, 116 F.3d 1454, 1457 (Fed. Cir. 1997). “Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim.” *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1458 (Fed.Cir. 1984). As stated in *Scripps Clinic & Research Foundation v. Genentech Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991), “anticipation requires that all of the elements and limitations of the claim are found within a single prior art reference. . . . There must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention.”

B. The Office Action Assertions and Applicants’ Response

Claims 1, 3-4, 6-7, 23-24 and 26-33 stand rejected under 35 U.S.C. §102.

The Office Action states:

Claims 1, 3-4, 6-7, 23-24, 26-33 are rejected under 35 U.S.C. 102(e) as being anticipated by Muller et al. (hereinafter “Muller”, 6,650,640 B1).

As per claim 1, Muller discloses an interface device for a computer, the interface device connectable to a network and a storage unit, the storage unit including a disk drive, the interface device comprising:

- A sequencer including a hardware logic circuit configured to process a transport layer header of a network packet (column 4, lines 48-50, column 7, lines 20-25, 31-35, 64-67, column 8, lines 1-5, 17-20, 50-60, column 9, lines 1-5, column 15, lines 35-38, column 35, lines 53-67, column 36, lines 11-30);
- A memory adapted to store control information regarding a network connection being handled by said device (column 4, lines 20-25, column 9, lines 14-16, 20-25, 56-58, column 10, lines 1-7, column 11, lines 46-59, column 12, lines 11-15, column 52, lines 64-67, column 53, lines 1-7);
- A mechanism for associating said packet with said control information and for selecting whether to process said packet by said computer or to send data from said packet to the storage unit,

thereby avoiding the computer (column 4, lines 45-50, 58-67, column 8, lines 50-60, 66-67, column 9, lines 13-17, 22-35, 66-67, column 10, lines 2-7, column 11, lines 46-59, column 12, lines 11-15, column 16, lines 59-67).

Applicants respectfully disagree with several of the foregoing Office Action statements. For instance, applicants respectfully assert that neither column 4, lines 45-50, 58-67, column 8, lines 50-60, 66-67, column 9, lines 13-17, 22-35, 66-67, column 10, lines 2-7, column 11, lines 46-59, column 12, lines 11-15, nor column 16, lines 59-67 of Muller discloses "A mechanism for associating said packet with said control information and for selecting whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer." Instead, column 4, lines 45-50 and 58-67 of Muller state:

When a flow packet is received at the network interface, a flow database manager receives the packet's flow key. The flow key may be assembled by a header parser module that parses a header portion of the packet. The flow database manager may also receive control information concerning the packet, such as an indication of the size of a data portion

...
manager associates an operation code with the received packet to indicate how the packet may be further processed by the network interface and/or a host computer. The specific operation code assigned for a packet may indicate whether the packet contains data that can be re-assembled with other data passed in the flow, whether the packet is a control packet or is otherwise devoid of data, whether the packet should not be processed through a particular network interface function (e.g., due to a flag in a header of the packet), etc.

Similarly, column 8, lines 50-60 and 66-67 of Muller state:

Header parser 106 parses a header portion of the packet to retrieve various pieces of information that will be used to identify related packets (e.g., multiple packets from one same source entity for one destination entity) and that will affect subsequent processing of the packets. In the illustrated embodiment, header parser 106 communicates with flow database manager (FDBM) 108, which manages flow database (FDB) 110. In particular, header parser 106 submits a query to FDBM 108 to determine whether a valid communication flow (described below) exists between the source entity that sent a packet and the destination entity. The destination entity may comprise an application program, a communication module, or some other element of a host computer system that is to receive the packet.

...

one source entity to one destination entity. A flow may be identified by a flow key assembled from source and desti-

Likewise, column 9, lines 13-17, 22-35 and 66-67 of Muller state:

Thus, for purposes of flow management, header parser 106 passes the packet's flow key to flow database manager 108. The header parser may also provide the flow database manager with other information concerning the packet that was retrieved from the packet (e.g., length of the packet).

...
entity served by NIC 100. Thus, FDBM 108 updates FDB 110 as necessary, depending upon the information received from header parser 106. In addition, in this embodiment of the invention FDBM 108 associates an operation or action code with the received packet. An operation code may be used to identify whether a packet is part of a new or existing flow, whether the packet includes data or just control information, the amount of data within the packet, whether the packet data can be re-assembled with related data (e.g., other data in a datagram sent from the source entity to the destination entity), etc. FDBM 108 may use information retrieved from the packet and provided by header parser 106 to select an appropriate operation code. The packet's operation code is then passed back to the header parser, along with

...
Now the packet may be stored in packet queue 116, which holds packets for manipulation by DMA (Direct Memory...

In addition, column 10, lines 2-7 of Muller state:

addition to storing the packet in a packet queue, a corresponding entry for the packet is made in control queue 118 and information concerning the packet's flow may also be passed to dynamic packet batching module 122. Control queue 118 contains related control information for each packet in packet queue 116.

Similarly, column 11, lines 46-59 of Muller state:

In state 136, information extracted from the packet's headers is forwarded to flow database manager 108 and/or load distributor 112. The FDBM uses the information to set up a flow in flow database 110 if one does not already exist for this communication flow. If an entry already exists for the packet's flow, it may be updated to reflect the receipt of a new flow packet. Further, FDBM 108 generates an operation code to summarize one or more characteristics or conditions of the packet. The operation code may be used by other modules of NIC 100 to handle the packet in an appropriate manner, as described in subsequent sections. The operation code is returned to the header parser, along with an index (e.g., a flow number) of the packet's flow in the flow database.

Continuing, column 12, lines 11-15 of Muller state:

Also in state 140, control information for the packet is stored in control queue 118 and information concerning the packet's flow (e.g., flow number, flow key) may be provided to dynamic packet batching module 122.

...

In state 142, NIC 100 determines whether the packet is...

And finally, column 16, lines 59-67 of Muller state:

The information generated by the header parser includes, in particular, a flow key with which to identify the communication flow or communication connection that comprises the received packet. In one embodiment of the invention, data from packets having the same flow key may be identified and re-assembled to form a datagram. In addition, headers of packets having the same flow key may be processed collectively through their protocol stack (e.g., rather than serially).

Applicants respectfully assert that none of the above-quoted passages that were cited by the Office Action teaches or suggests “a mechanism for associating said packet with said control information and for selecting whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer,” as recited in claim 1. For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 1 or any claim that depends from claim 1.

Regarding claim 3, the Office Action states:

As per claim 3, Muller discloses the interface device of claim 1, further comprising a plurality of network ports, wherein one of the said network ports is connectable to a storage unit (column 4, lines 40-43, column 6, lines 37-40, column 7, lines 15-19, column 8, lines 40-43, column 9, lines 1-5, column 10, lines 65-67).

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 3 and respectfully assert that none of those passages teaches or suggests “The interface device of claim 1, further comprising a plurality of network ports, wherein one of said network ports is connectable to the storage unit.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 3.

Regarding claim 4, the Office Action states:

As per claim 4, Muller discloses the interface device of claim 1, further comprising a Fibre Channel controller connectable to the storage unit (column 61, lines 55-60).

Instead, column 61, lines 55-60 of Muller states:

Reserving sixty-four bytes at the beginning of a buffer also allows header information to be modified or prepended if necessary. For example, a regular Ethernet header of a packet may, because of routing requirements, need to be replaced with a much larger FDDI (Fiber Distributed Data Interface) header. One skilled in the art will recognize the...

Applicants respectfully assert that the above-quoted passage that was cited by the Office Action does not teach or suggest “the interface device of claim 1, further comprising a Fibre Channel controller connectable to the storage unit,” as recited in claim 4. For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 4.

Regarding claim 6, the Office Action states:

As per claim 6, Muller discloses the network interface device of claim 1, further comprising a file cache adapted to store said data (column 56, lines 20-30, column 58, lines 26-30, column 61, lines 34-35, column 62, lines 47-52).

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 6. Without reprinting each of those passages here, suffice it to say that applicants respectfully assert that none of those passages teaches or suggests “the interface device of claim 1, further comprising a file cache adapted to store said data.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 6.

Regarding claim 7, the Office Action states:

As per claim 7, Muller further discloses the interface device of claim 1, further comprising a file cache adapted to store said data under control of a file system in the host (column 56, lines 20-30, column 58, lines 26-30, column 61, lines 34-35, column 62, lines 47-52).

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 7 and respectfully assert that none of those passages teaches or suggests “The interface device of claim 1, further comprising a file cache adapted to store said

data under control of a file system in the host.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 7.

Regarding claim 21, the Office Action states:

As per claim 21, Muller discloses an interface device for a computer, the interface device connectable to a network and a storage unit, the storage unit including a disk drive, the interface device comprising:

- A receive mechanism that processes a Transmission Control Protocol (TCP) header of a network packet (column 4, lines 48-50, column 7, lines 20-25, 31-35, 64-67, column 8, lines 1-5, 17-20, 50-60, column 9, lines 1-5, column 15, lines 35-38, column 35, lines 53-67, column 36, lines 11-30);
- A memory storing a combination of information describing an established TCP connection (column 4, lines 20-25, column 9, lines 14-16, 20-25, 56-58, column 10, lines 1-7, column 11, lines 46-59, column 12, lines 11-15, column 52, lines 64-67, column 53, lines 1-7);
- A processing mechanism that associates said packet with said information and selects whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer (column 4, lines 45-50, 58-67, column 8, lines 50-60, 66-67, column 9, lines 13-17, 22-35, 66-67, column 10, lines 2-7, column 11, lines 46-59, column 12, lines 11-15, column 16, lines 59-67).

Applicants respectfully disagree with several of the foregoing Office Action statements. For instance, applicants respectfully assert that neither column 4, lines 45-50, 58-67, column 8, lines 50-60, 66-67, column 9, lines 13-17, 22-35, 66-67, column 10, lines 2-7, column 11, lines 46-59, column 12, lines 11-15, nor column 16, lines 59-67 of Muller discloses “A processing mechanism that associates said packet with said information and selects whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer.” Instead, column 4, lines 45-50 and 58-67 of Muller state:

When a flow packet is received at the network interface, a flow database manager receives the packet's flow key. The flow key may be assembled by a header parser module that parses a header portion of the packet. The flow database manager may also receive control information concerning the packet, such as an indication of the size of a data portion

...
manager associates an operation code with the received packet to indicate how the packet may be further processed by the network interface and/or a host computer. The specific operation code assigned for a packet may

indicate whether the packet contains data that can be re-assembled with other data passed in the flow, whether the packet is a control packet or is otherwise devoid of data, whether the packet should not be processed through a particular network interface function (e.g., due to a flag in a header of the packet), etc.

Similarly, column 8, lines 50-60 and 66-67 of Muller state:

Header parser 106 parses a header portion of the packet to retrieve various pieces of information that will be used to identify related packets (e.g., multiple packets from one same source entity for one destination entity) and that will affect subsequent processing of the packets. In the illustrated embodiment, header parser 106 communicates with flow database manager (FDBM) 108, which manages flow database (FDB) 110. In particular, header parser 106 submits a query to FDBM 108 to determine whether a valid communication flow (described below) exists between the source entity that sent a packet and the destination entity. The destination entity may comprise an application program, a communication module, or some other element of a host computer system that is to receive the packet.

...

one source entity to one destination entity. A flow may be identified by a flow key assembled from source and desti-

Likewise, column 9, lines 13-17, 22-35 and 66-67 of Muller state:

Thus, for purposes of flow management, header parser 106 passes the packet's flow key to flow database manager 108. The header parser may also provide the flow database manager with other information concerning the packet that was retrieved from the packet (e.g., length of the packet).

...

entity served by NIC 100. Thus, FDBM 108 updates FDB 110 as necessary, depending upon the information received from header parser 106. In addition, in this embodiment of the invention FDBM 108 associates an operation or action code with the received packet. An operation code may be used to identify whether a packet is part of a new or existing flow, whether the packet includes data or just control information, the amount of data within the packet, whether the packet data can be re-assembled with related data (e.g., other data in a datagram sent from the source entity to the destination entity), etc. FDBM 108 may use information retrieved from the packet and provided by header parser 106 to select an appropriate operation code. The packet's operation code is then passed back to the header parser, along with

...

Now the packet may be stored in packet queue 116, which holds packets for manipulation by DMA (Direct Memory...

In addition, column 10, lines 2-7 of Muller state:

addition to storing the packet in a packet queue, a corresponding entry for the packet is made in control queue 118 and information concerning the packet's flow may also be passed to dynamic packet batching module 122. Control queue 118 contains related control information for each packet in packet queue 116.

Similarly, column 11, lines 46-59 of Muller state:

In state 136, information extracted from the packet's headers is forwarded to flow database manager 108 and/or load distributor 112. The FDBM uses the information to set up a flow in flow database 110 if one does not already exist for this communication flow. If an entry already exists for the packet's flow, it may be updated to reflect the receipt of a new flow packet. Further, FDBM 108 generates an operation code to summarize one or more characteristics or conditions of the packet. The operation code may be used by other modules of NIC 100 to handle the packet in an appropriate manner, as described in subsequent sections. The operation code is returned to the header parser, along with an index (e.g., a flow number) of the packet's flow in the flow database.

Continuing, column 12, lines 11-15 of Muller state:

Also in state 140, control information for the packet is stored in control queue 118 and information concerning the packet's flow (e.g., flow number, flow key) may be provided to dynamic packet batching module 122.

...

In state 142, NIC 100 determines whether the packet is...

And finally, column 16, lines 59-67 of Muller state:

The information generated by the header parser includes, in particular, a flow key with which to identify the communication flow or communication connection that comprises the received packet. In one embodiment of the invention, data from packets having the same flow key may be identified and re-assembled to form a datagram. In addition, headers of packets having the same flow key may be processed collectively through their protocol stack (e.g., rather than serially).

Applicants respectfully assert that none of the above-quoted passages that were cited by the Office Action teaches or suggests "a mechanism for associating said packet with said control information and for selecting whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the

computer,” as recited in claim 21. For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 21 or any claim that depends from claim 21.

Regarding claim 23, the Office Action states:

As per claim 23, Muller discloses the interface device of claim 21, further comprising a Fibre Channel controller connectable to the storage unit (column 61, lines 55-60).

Applicants initially note that claim 23 recites:

The interface device of claim 21, further comprising a plurality of network ports, wherein one of said network ports is connectable to the storage unit.

In contrast, column 61, lines 55-60 of Muller recite:

Reserving sixty-four bytes at the beginning of a buffer also allows header information to be modified or prepended if necessary. For example, a regular Ethernet header of a packet may, because of routing requirements, need to be replaced with a much larger FDDI (Fiber Distributed Data Interface) header. One skilled in the art will recognize the...

Applicants respectfully assert that the above-quoted passage that was cited by the Office Action does not teach or suggest “the interface device of claim 21, further comprising a plurality of network ports, wherein one of said network ports is connectable to the storage unit,” as recited in claim 23. For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 23.

Regarding claim 24, the Office Action states:

As per claim 24, Muller discloses the network interface device of claim 1, further comprising a file cache adapted to store said data (column 56, lines 20-30, column 58, lines 26-30, column 61, lines 34-35, column 62, lines 47-52).

Applicants initially note that claim 24 recites:

The interface device of claim 21, further comprising a Fibre Channel controller connectable to the storage unit.

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 24 and respectfully assert that none of those passages teaches or suggests “the interface device of claim 1, further comprising a Fibre Channel controller

connectable to the storage unit.” Nor does the passage of Muller cited by the Office Action with regard to claim 23 (column 61, lines 55-60) teach or suggest the limitations of claim 24. For at least these reasons, the Office Action has not presented a prima facie case of anticipation for claim 24.

Regarding claim 26, the Office Action states:

As per claim 26, Muller discloses the network interface device of claim 21, further comprising a file cache adapted to store said data (column 56, lines 20-30, column 58, lines 26-30, column 61, lines 34-35, column 62, lines 47-52).

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 26. Without reprinting each of those passages here, suffice it to say that applicants respectfully assert that none of those passages teaches or suggests “the interface device of claim 21, further comprising a file cache adapted to store said data.” Instead, some of those passages teach a cache for descriptors. For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 26.

Regarding claim 27, the Office Action states:

As per claim 27, Muller further discloses the interface device of claim 21, further comprising a file cache adapted to store said data under control of a file system in the host (column 56, lines 20-30, column 58, lines 26-30, column 61, lines 34-35, column 62, lines 47-52).

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 27 and respectfully assert that none of those passages teaches or suggests “The interface device of claim 21, further comprising a file cache adapted to store said data under control of a file system in the host.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 27.

Regarding claim 28, the Office Action states:

As per claim 28, Muller discloses a method for operating an interface device for a computer, the interface device connectable to a network and a storage unit, the storage unit including a disk drive, the method comprising:

- Receiving, by the interface device from the network, a packet containing data and a Transmission Control Protocol (TCP) header (column 4, lines 48-50, column 7, lines 20-25, 31-35, 64-67, column 8, lines 1-5, 17-20, 50-60, column 9, lines 1-5, column 15, lines 35-38, column 35, lines 53-67, column 36, lines 11-30);

- Processing, by the interface device, the TCP header (column 4, lines 45-50, 58-67, column 8, lines 50-60, column 9, lines 13-17, 22-35, 66-67);
- Storing, on the interface device, information regarding a TCP connection (A memory adapted to store control information regarding a network connection being handled by said device (column 4, lines 20-25, column 9, lines 14-16, 20-25, 56-58, column 10, lines 1-7, column 11, lines 46-59, column 12, lines 11-15);
- Selecting, by the interface device, whether to process the packet by the computer or to send the data from the packet to the storage unit, thereby avoiding the computer (column 4, lines 45-50, 58-67, column 8, lines 50-60, 66-67, column 9, lines 13-17, 22-35, 66-67, column 10, lines 2-7, column 11, lines 46-59, column 12, lines 11-15, column 16, lines 59-67).

Applicants respectfully disagree with several of the foregoing Office Action statements. For instance, applicants respectfully assert that neither column 4, lines 45-50, 58-67, column 8, lines 50-60, 66-67, column 9, lines 13-17, 22-35, 66-67, column 10, lines 2-7, column 11, lines 46-59, column 12, lines 11-15, nor column 16, lines 59-67 of Muller discloses "Selecting, by the interface device, whether to process the packet by the computer or to send the data from the packet to the storage unit, thereby avoiding the computer." Instead, column 4, lines 45-50 and 58-67 of Muller state:

When a flow packet is received at the network interface, a flow database manager receives the packet's flow key. The flow key may be assembled by a header parser module that parses a header portion of the packet. The flow database manager may also receive control information concerning the packet, such as an indication of the size of a data portion

... manager associates an operation code with the received packet to indicate how the packet may be further processed by the network interface and/or a host computer. The specific operation code assigned for a packet may indicate whether the packet contains data that can be re-assembled with other data passed in the flow, whether the packet is a control packet or is otherwise devoid of data, whether the packet should not be processed through a particular network interface function (e.g., due to a flag in a header of the packet), etc.

Similarly, column 8, lines 50-60 and 66-67 of Muller state:

Header parser 106 parses a header portion of the packet to retrieve various pieces of information that will be used to identify related packets (e.g., multiple packets from one same source entity for one destination

entity) and that will affect subsequent processing of the packets. In the illustrated embodiment, header parser 106 communicates with flow database manager (FDBM) 108, which manages flow database (FDB) 110. In particular, header parser 106 submits a query to FDBM 108 to determine whether a valid communication flow (described below) exists between the source entity that sent a packet and the destination entity. The destination entity may comprise an application program, a communication module, or some other element of a host computer system that is to receive the packet.

...

one source entity to one destination entity. A flow may be identified by a flow key assembled from source and desti-

Likewise, column 9, lines 13-17, 22-35 and 66-67 of Muller state:

Thus, for purposes of flow management, header parser 106 passes the packet's flow key to flow database manager 108. The header parser may also provide the flow database manager with other information concerning the packet that was retrieved from the packet (e.g., length of the packet).

...

entity served by NIC 100. Thus, FDBM 108 updates FDB 110 as necessary, depending upon the information received from header parser 106. In addition, in this embodiment of the invention FDBM 108 associates an operation or action code with the received packet. An operation code may be used to identify whether a packet is part of a new or existing flow, whether the packet includes data or just control information, the amount of data within the packet, whether the packet data can be re-assembled with related data (e.g., other data in a datagram sent from the source entity to the destination entity), etc. FDBM 108 may use information retrieved from the packet and provided by header parser 106 to select an appropriate operation code. The packet's operation code is then passed back to the header parser, along with

...

Now the packet may be stored in packet queue 116, which holds packets for manipulation by DMA (Direct Memory...

In addition, column 10, lines 2-7 of Muller state:

addition to storing the packet in a packet queue, a corresponding entry for the packet is made in control queue 118 and information concerning the packet's flow may also be passed to dynamic packet batching module 122. Control queue 118 contains related control information for each packet in packet queue 116.

Similarly, column 11, lines 46-59 of Muller state:

In state 136, information extracted from the packet's headers is forwarded to flow database manager 108 and/or load distributor 112. The FDBM uses the information to set up a flow in flow database 110 if one does not already exist for this communication flow. If an entry already exists for the packet's flow, it may be updated to reflect the receipt of a new flow packet. Further, FDBM 108 generates an operation code to summarize one or more characteristics or conditions of the packet. The operation code may be used by other modules of NIC 100 to handle the packet in an appropriate manner, as described in subsequent sections. The operation code is returned to the header parser, along with an index (e.g., a flow number) of the packet's flow in the flow database.

Continuing, column 12, lines 11-15 of Muller state:

Also in state 140, control information for the packet is stored in control queue 118 and information concerning the packet's flow (e.g., flow number, flow key) may be provided to dynamic packet batching module 122.

...

In state 142, NIC 100 determines whether the packet is...

And finally, column 16, lines 59-67 of Muller state:

The information generated by the header parser includes, in particular, a flow key with which to identify the communication flow or communication connection that comprises the received packet. In one embodiment of the invention, data from packets having the same flow key may be identified and re-assembled to form a datagram. In addition, headers of packets having the same flow key may be processed collectively through their protocol stack (e.g., rather than serially).

Applicants respectfully assert that none of the above-quoted passages that were cited by the Office Action teaches or suggests "Selecting, by the interface device, whether to process the packet by said computer or to send the data from the packet to the storage unit, thereby avoiding the computer," as recited in claim 28. For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 28 or any claim that depends from claim 28.

Regarding claim 29, the Office Action states:

As per claim 29, Muller discloses the method of claim 28, further comprising creating, by the computer, the information regarding the TCP connection (column 5, lines 35-45).

Applicants have reviewed the passage from Muller that the Office Action cites regarding claim 29, and respectfully assert that column 5, lines 35-45 does not teach or suggest “the method of claim 28, further comprising creating, by the computer, the information regarding the TCP connection.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 29.

Regarding claim 30, the Office Action states:

As per claim 30, Muller discloses the method of claim 28, wherein the interface device includes a network port, and the packet is received via the port and the data is sent to the storage unit via the port (column 10, lines 1-7).

Applicants have reviewed the passage from Muller that the Office Action cites regarding claim 30, and respectfully assert that column 10, lines 1-7 do not teach or suggest “the method of claim 28, wherein the interface device includes a network port, and the packet is received via the port and the data is sent to the storage unit via the port.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 30.

Regarding claim 31, the Office Action states:

As per claim 31, Muller discloses the method of claim 28, wherein the interface device includes first and second network ports, and the packet is received via the first port and the data is sent to the storage unit via the second port (column 10, lines 35-47).

Applicants have reviewed the passage from Muller that the Office Action cites regarding claim 31, and respectfully assert that column 10, lines 35-47 do not teach or suggest “the method of claim 28, wherein the interface device includes first and second network ports, and the packet is received via the first port and the data is sent to the storage unit via the second port.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 31.

Regarding claim 32, the Office Action states:

As per claim 32, Muller discloses the method of claim 28, further comprising storing the data on a file cache of the interface device (column 56, lines 20-30, column 58, lines 26-30, column 61, lines 34-35, column 62, lines 47-52).

Applicants have reviewed the passages from Muller that the Office Action cites regarding claim 32, and respectfully assert that the passages do not teach or suggest “the method of claim 28, further comprising storing the data on a file cache of the interface device.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 32.

Regarding claim 33, the Office Action states:

As per claim 33, discloses the method of claim 28, further comprising adding a network protocol header to the data for sending the data to the storage unit (column 9, lines 50-67).

Applicants have reviewed the passage from Muller that the Office Action cites regarding claim 33, and respectfully assert that this passage does not teach or suggest “the method of claim 28, further comprising adding a network protocol header to the data for sending the data to the storage unit.” For at least this reason, the Office Action has not presented a prima facie case of anticipation for claim 33.

II. 35 U.S.C. §103

A. The Standard

“The combination of elements from non-analogous sources, in a manner that reconstructs the applicant's invention only with the benefit of hindsight, is insufficient to present a prima facie case of obviousness. There must be some reason, suggestion, or motivation found in the prior art whereby a person of ordinary skill in the field of the invention would make the combination. That knowledge can not come from the applicant's invention itself.” *In re Oetiker*, 24 USPQ 2d 1443, 1446 (Fed. Cir. 1992). See also *In re Clay*, 966 F.2d 656, 658-660 (Fed. Cir. 1992) and *Monarch Knitting Mach. Corp. v. Sulzer Morat GmbH*, 139 F.3d 877, 881 (Fed. Cir., 1998). “Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. Under section 103, teachings of references can be combined *only* if there is some suggestion or incentive to do so... The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art suggested the desirability of the modification.” *In re Fritch*, 972 F.2d 1260, 1266 (Fed. Cir. 1992). “In proceedings before

the Patent and Trademark Office, the Examiner bears the burden of establishing a prima facie case of obviousness based upon the prior art.” *Id.* at 1265.

B. The Office Action Assertions and Applicants’ Response

Claims 2, 5, 17, 22 and 25 stand rejected under 35 U.S.C. §103(a). Note, however, that claim 17 was canceled in the Election and Amendment dated July 1, 2005. The Office Action states:

Claims 2, 5, 17, 22, 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muller et al. (hereinafter “Muller”, 6,650,640 B1) in view of Day et al. (hereinafter “Day”, 6,065,096).

Regarding claims 2, 17 and 22, the Office Action states:

As per claims 2, 17 and 22, Muller discloses the interface device of claims 1 and 21.

Muller does not explicitly disclose the interface further comprising a SCSI controller connectable to the storage unit.

However, Day discloses SCSI interface channels attached to disk drives (column 2, lines 40-54, column 5, lines 1-25).

Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to implement or incorporate in Muller’s device Day’s interface comprising a SCSI controller in order to provide for a simple, lower cost RAID controller architecture to enable lower cost and complexity associated with high performance and high reliability storage subsystems.

Applicants respectfully assert that, as noted above regarding claim 1, Muller does not teach or suggest “a mechanism for associating said packet with said control information and for selecting whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer.” Similarly, as noted above regarding claim 21, Muller does not teach or suggest “a processing mechanism that associates said packet with said information and selects whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer.” Applicants respectfully assert that Day also does not teach or suggest either of these limitations, and that implementing or incorporating in Muller’s device Day’s interface as proposed by the Office Action would not solve this deficiency.

Moreover, the motivation to make this modification is contradicted by the reasoning cited by the Office Action. Somehow implementing or incorporating in Muller's device Day's interface would increase the cost and complexity of Muller's device without any obvious benefit. Muller's device "relates to a network interface circuit (NIC) for processing communication packets exchanged between a computer network and a host computer system" (column 1, lines 51-54). Day's RAID controller, on the other hand, provides a chip for controlling a redundant array of inexpensive disk drives (column 1, lines 11-19; column 2, lines 11-24). Stated differently, Muller involves network communication and Day involves storage. Applicants respectfully assert that, absent the teachings of the present invention, no motivation is apparent in the cited references to make the modification proposed by the Office Action.

Regarding claims 5 and 25, the Office Action states:

As per claims 5 and 25, Muller discloses the interface device of claims 1 and 21.

Muller does not explicitly disclose the interface further comprising a RAID controller connectable to the storage unit.

However, Day discloses a RAID controller that integrates onto a single integrated circuit of a general purpose processor (column 2, lines 11-25, 55-67).

Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to implement or incorporate in Muller's device Day's interface comprising a RAID controller allowing the disk interface connections and protocols to be more flexibly selected but at the cost of less integration within the circuit.

As noted above regarding claim 1, Muller does not teach or suggest "a mechanism for associating said packet with said control information and for selecting whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer." Similarly, as noted above regarding claim 21, Muller does not teach or suggest "a processing mechanism that associates said packet with said information and selects whether to process said packet by said computer or to send data from said packet to the storage unit, thereby avoiding the computer." Day also does not teach or suggest either of these limitations. Implementing or incorporating in Muller's device Day's interface as proposed by the Office Action would not solve this deficiency.

Moreover, the motivation asserted by the Office Action to make this modification is quoted from Day at column 2, lines 51-54, which refers to an alternative storage configuration, not anything involving network communication. As mentioned above, somehow implementing or incorporating in Muller's device Day's interface would increase the cost and complexity of Muller's device without any obvious benefit. Muller's device "relates to a network interface circuit (NIC) for processing communication packets exchanged between a computer network and a host computer system" (column 1, lines 51-54). Day's RAID controller, on the other hand, provides a chip for controlling a redundant array of inexpensive disk drives (column 1, lines 11-19; column 2, lines 11-24). Stated differently, Muller involves network communication and Day involves storage. Absent the teachings of the present invention, no motivation is apparent in the cited references to make the modification proposed by the Office Action.

III. Conclusion

In this Amendment, applicants have responded to each item of the Office Action by explaining why the Office Action does not present a prima facie case of anticipation or obviousness for any of the claims. As such, applicants respectfully assert that the application is in condition for allowance, and a notice of allowance is solicited.

Respectfully submitted,



Mark Lauer
Reg. No. 36,578
6601 Koll Center Parkway
Suite 245
Pleasanton, CA 94566
Tel: (925) 484-9295
Fax: (925) 484-9291

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on December 19, 2005.

Date: 12-19-05


Mark Lauer